

使用 MapKit 绘制路线

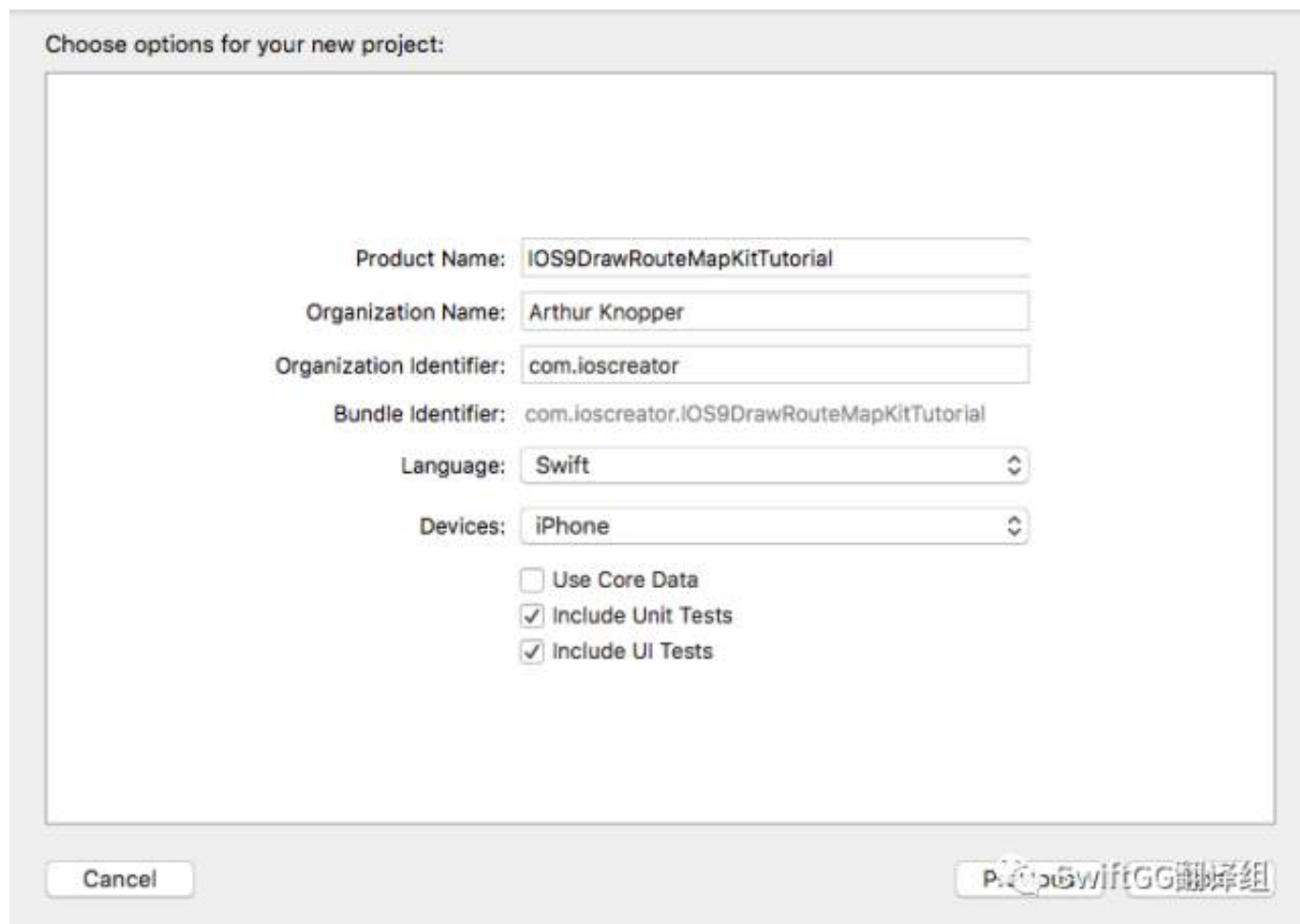
原创 2017-05-02 Arthur SwiftGG翻译组

译者：TonyHan；校对：Crystal Sun；定稿：CMB

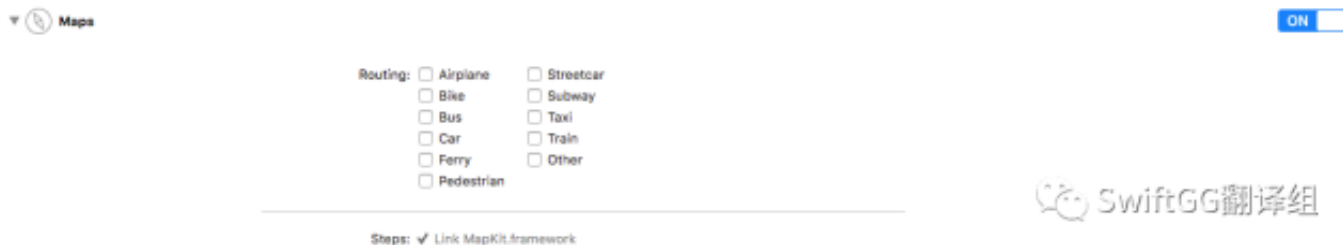
在本节教程中，将学习如何在纽约市的两个地标之间绘制一条线路。这条线路绘制在 Map 图层的上面，使用折线绘制。本教程使用 Xcode 7.2 和 iOS 9.2。

译者注：由于目前主流版本为 Xcode 8 (iOS 10)，因此图示可能会有所不同，不过不影响对于文章的理解。基于此版本的项目可以在[这里](#)下载到。

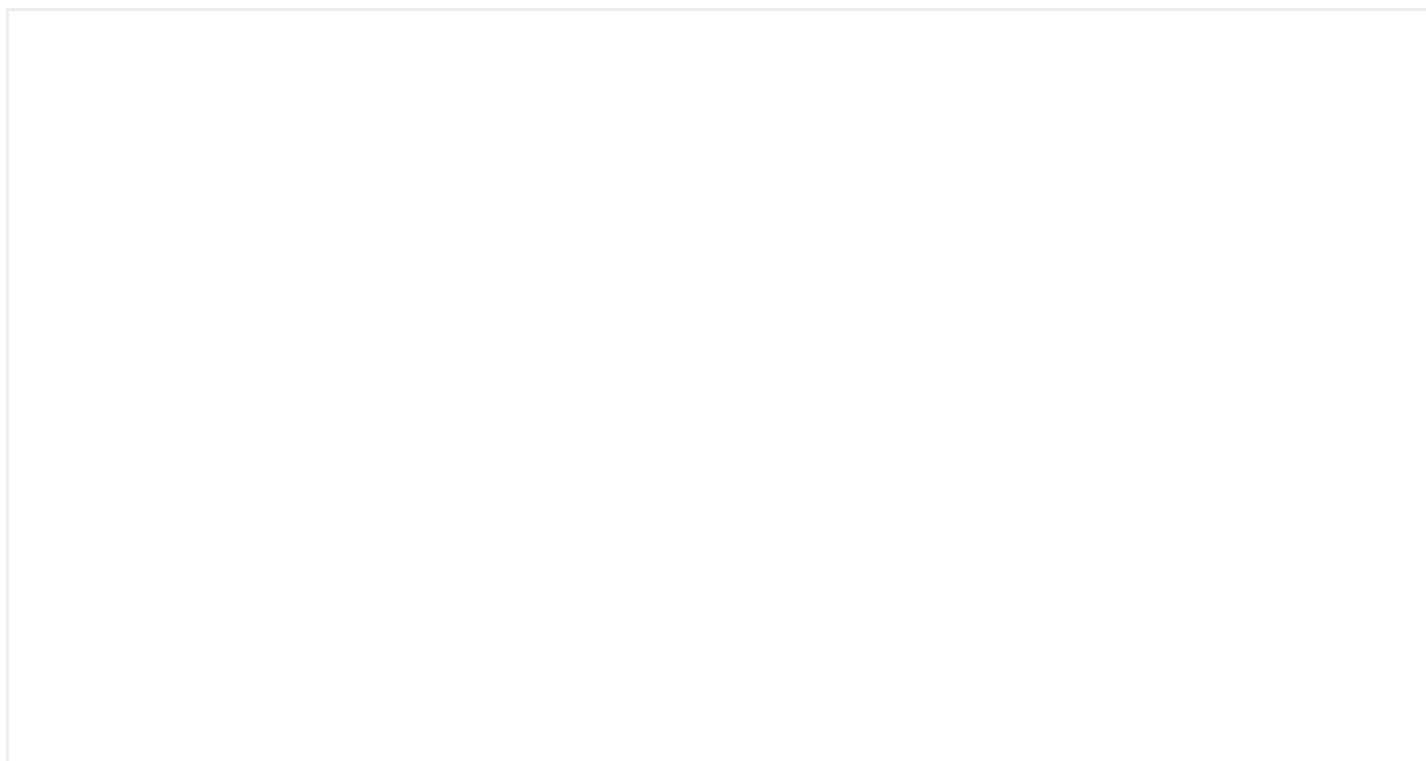
打开 Xcode 创建 Single View Application。使用 **IOS9DrawRouteMapKitTutorial** 作为项目名称，然后根据实际情况填写 Organization Name 和 Organization Identifier。选择 Swift 作为编程语言，Devices 选择只适配 iPhone。



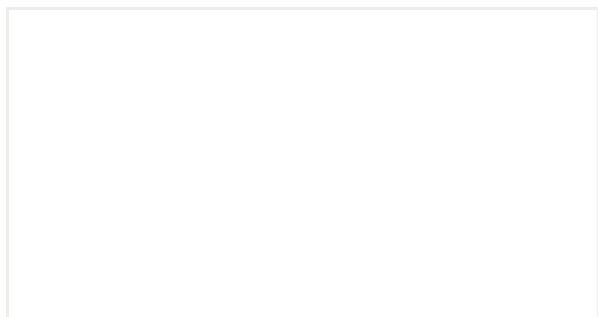
在项目导航栏中，选择 Project Settings 图标。点击 Capabilities 栏并启用 Map framework。



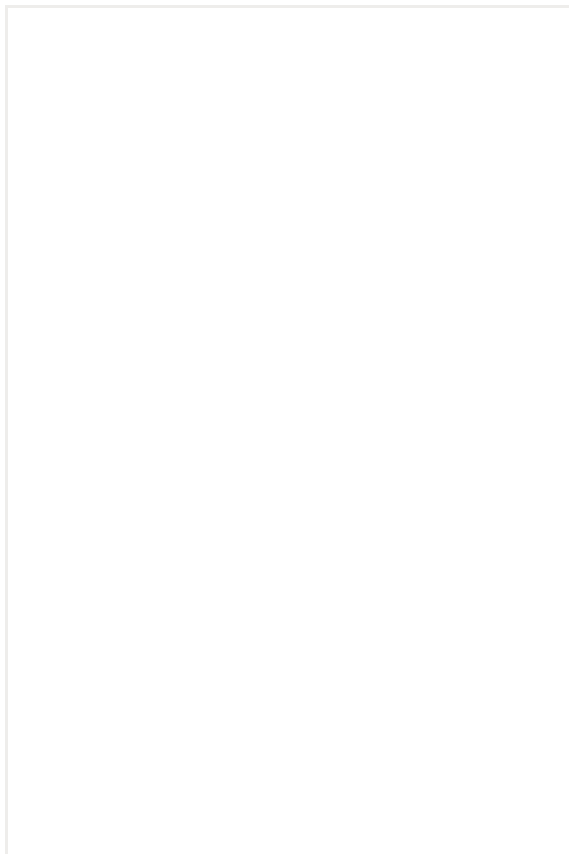
打开 **Storyboard**，选择 View Controller，打开 Editor 菜单并选择 Embed in -> Navigation Controller。双击 View Controller 的 Navigation Bar 并输入“Route Tutorial”。接下来，从视图对象库中拖拽 Map Kit View 到主视图中。重新设置尺寸，让其填充整个屏幕。现在 Storyboard 如下图所示：



打开 Assistant Editor，确保 **ViewController.swift** 文件可见。按住 control 键将 Map Kit View 拖动到 ViewController 类中，创建如下图所示的 Outlet：



选择 Map Kit View，并点击在 Interface Builder 右下角的 Pin 图标（左边起第三个）。四边都设置约束，并反选“Constrain to margins”。点击“Add 4 Constraints”按钮，于是 Map Kit View 在各方向上边缘都与主视图对齐。



打开 Assistant Editor，确保 ViewController.swift 文件可见。选择 Map Kit View。按住 control 键将 Map Kit View 拖动到 ViewController 类中，创建的 Outlet。

译者注：这一段内容与上面的创建 Outlet 内容有重复，上面那一段及后面的图片应该在此处。因原文如此，故在此未改。

打开 **ViewController.swift**，添加 MapKit framework。

```
import MapKit
```

View Controller 需要遵循 MKMapViewDelegate 协议，才能实现路线的绘制。将类声明的代码修改为：

```
class ViewController: UIViewController, MKMapViewDelegate {
```

然后，将 **viewDidLoad** 方法修改为：

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    // 1.  
    mapView.delegate = self
```

```
// 2.
let sourceLocation = CLLocationCoordinate2D(latitude: 40.759011, longitude: -73.984472)
let destinationLocation = CLLocationCoordinate2D(latitude: 40.748441, longitude: -73.985564)

// 3.
let sourcePlacemark = MKPlacemark(coordinate: sourceLocation, addressDictionary: nil)
let destinationPlacemark = MKPlacemark(coordinate: destinationLocation, addressDictionary: nil)

// 4.
let sourceMapItem = MKMapItem(placemark: sourcePlacemark)
let destinationMapItem = MKMapItem(placemark: destinationPlacemark)

// 5.
let sourceAnnotation = MKPointAnnotation()
sourceAnnotation.title = "Times Square"

if let location = sourcePlacemark.location {
    sourceAnnotation.coordinate = location.coordinate
}

let destinationAnnotation = MKPointAnnotation()
destinationAnnotation.title = "Empire State Building"

if let location = destinationPlacemark.location {
    destinationAnnotation.coordinate = location.coordinate
}

// 6.
self.mapView.showAnnotations([sourceAnnotation, destinationAnnotation], animated: true )

// 7.
let directionRequest = MKDirectionsRequest()
directionRequest.source = sourceMapItem
directionRequest.destination = destinationMapItem
directionRequest.transportType = .Automobile

// 计算方向
let directions = MKDirections(request: directionRequest)

// 8.
directions.calculateDirectionsWithCompletionHandler {
```

```

        (response, error) -> Void in

        guard let response = response else {
            if let error = error {
                print("Error: \(error)")
            }

            return
        }

        let route = response.routes[0]
        self.mapView.addOverlay((route.polyline), level: MKOverlayLevel.AboveRoads)

        let rect = route.polyline.boundingBox
        self.mapView.setRegion(MKCoordinateRegionForMapRect(rect), animated: true)
    }
}

```

1. ViewController 作为 MKMapViewDelegate 协议的代理
2. 设置地点的经纬度
3. 创建包含地点坐标的地标对象
4. 使用 MKMapItems 标记路径。该类封装了有关地图上特定点的信息
5. 添加显示地标名字的大头针
6. 在地图上显示大头针
7. 使用 MKDirectionsRequest 类计算路径
8. 使用折线在地图图层上面绘制出路径。该区域设置为两个位置都可见

接下来，实现代理方法 `mapView(rendererForOverlay:)`：

```

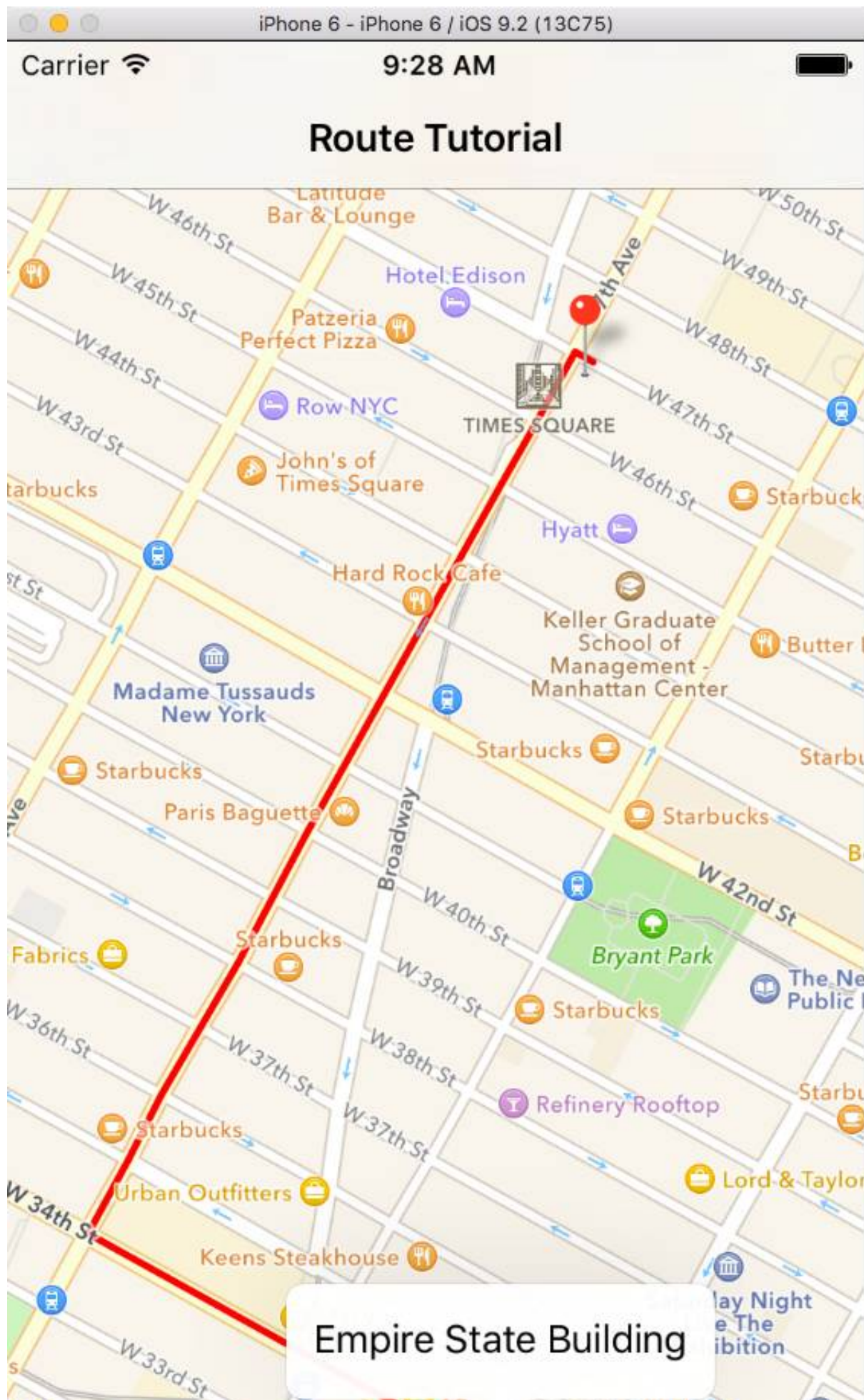
func mapView(mapView: MKMapView, rendererForOverlay overlay: MKOverlay) -> MKOverlayRenderer {
    let renderer = MKPolylineRenderer(overlay: overlay)
    renderer.strokeColor = UIColor.redColor()
    renderer.lineWidth = 4.0

    return renderer
}

```

这个方法返回用于返回绘制在地图上的渲染对象。使用宽度为 4 的红色线条。

运行 项目，展示出两个地点及其之间的路线。





可以从 Github 上 ioscreator 的仓库中下载 **IOS9DrawRouteMapKitTutorial** 项目的源代码作为参考。

译者注：由于目前主流版本为 Xcode 8 (iOS 10)，因此图示可能会有所不同，不过不影响对于文章的理解。基于此版本的项目可以在[这里](#)下载到。

本文由 SwiftGG 翻译组翻译，已经获得作者翻译授权，最新文章请访问 <http://swift.gg>。

[阅读原文](#)